

(19) 日本国特許庁 ( J P )

(12) 公 開 特 許 公 報 ( A )

(11) 特許出願公開番号

特開平8-263283

(43) 公開日 平成 8 年 (1996) 10 月 11 日

| (51) Int.Cl. <sup>6</sup> | 識別記号  | 庁内整理番号  | F I          | 技術表示箇所  |
|---------------------------|-------|---------|--------------|---------|
| G 0 6 F 9/06              | 5 5 0 |         | G 0 6 F 9/06 | 5 5 0 G |
| 15/00                     | 3 3 0 | 9364-5L | 15/00        | 3 3 0 D |

審査請求 未請求 請求項の数 3 O L (全 11 頁)

(21) 出願番号 特願平7-66081

(22) 出願日 平成 7 年 (1995) 3 月 24 日

(71) 出願人 000006747

株式会社リコー

東京都大田区中馬込 1 丁目 3 番 6 号

(72) 発明者 小林 秀樹

東京都大田区中馬込 1 丁目 3 番 6 号 株式  
会社リコー内

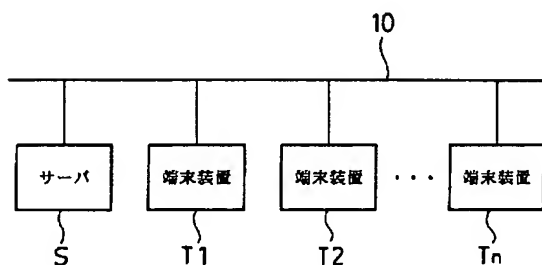
(74) 代理人 弁理士 大澤 敬

(54) 【発明の名称】 ソフトウェア管理システム

(57) 【要約】

【目的】 L A N 等の通信手段で接続した複数の端末装置でソフトウェアのプログラムが不正使用されないように容易に保守及び管理できるようにする。

【構成】 端末装置 T n がプログラムの実行要求がある度にそのプログラム識別名と端末装置識別名とユーザ識別名等のセキュリティチェック情報をサーバ S へ送信して、サーバ S が予め記憶したプログラム識別名とそれを使用可能な端末装置識別名とユーザ識別名等の管理情報と端末装置 T n から受信したセキュリティチェック情報とに基づいてその端末装置 T n でのプログラムの使用可否の妥当性を判断して、その判断結果をセキュリティチェック情報を送信してきた端末装置 T n へ送信して、端末装置 T n がサーバ S から受信した判断結果に基づいてプログラムに対する実行権を付与する。



## 【特許請求の範囲】

【請求項1】 サーバと複数の端末装置とを通信手段で接続し、該各端末装置によってプログラムを共用するソフトウェア管理システムにおいて、

前記サーバに、前記プログラムの識別名と該プログラムを使用可能な端末装置の識別名とユーザの識別名と該ユーザのパスワード等の管理情報を記憶する管理情報記憶手段と、該手段に記憶された管理情報と前記端末装置から受信したセキュリティチェック情報とに基づいて該端末装置での前記プログラムの使用可否の妥当性を判断する手段と、該手段による判断結果を前記セキュリティチェック情報を送信してきた端末装置へ送信する手段とを設け、

前記端末装置に、前記プログラムの実行要求がある度に該プログラムの識別名と端末装置の識別名とユーザの識別名と該ユーザのパスワード等のセキュリティチェック情報を前記サーバへ送信する手段と、前記サーバから受信した判断結果に基づいて前記プログラムに対する実行権を付与する手段とを設けたことを特徴とするソフトウェア管理システム。

【請求項2】 請求項1記載のソフトウェア管理システムにおいて、

前記サーバに、前記端末装置における前記プログラムの実行状態を定期的に問い合わせ、該端末装置から返送されるプログラムの現在の実行状態に基づいてそのプログラムの実行状態を監視する手段を設け、

前記端末装置に、前記サーバからの問い合わせがあったときに前記プログラムの現在の実行状態を返送する手段を設けたことを特徴とするソフトウェア管理システム。

【請求項3】 請求項1記載のソフトウェア管理システムにおいて、

前記サーバに、前記端末装置から前記プログラムに対して予め決められた同時実行可能数を越える実行要求を受け付けたとき、及び前記プログラムが予め決められた同時実行可能数で実行されているときに新たな実行要求を受け付けたとき、該実行要求を予め設定した優先度によって調停する手段を設けたことを特徴とするソフトウェア管理システム。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】 この発明は、サーバと複数の端末装置とをローカルエリアネットワーク（LAN）等の通信手段で接続し、その各端末装置によって各種のプログラムを共用するソフトウェア管理システムに関する。

## 【0002】

【従来の技術】 現在、コンピュータのプログラムを使用する場合、使用者はプログラムの販売者との間で契約を取り交わしてそのプログラムを使用する権利（「ライセンス」と称する）を得て使用することが求められている。そして、プログラムにそれを使用するコンピュータ

とユーザと同時使用数などの使用条件を課して、その使用条件以外ではプログラムの使用を認めないようにして、プログラムの不正使用を防止している。

【0003】 また、許可なくしてプログラムを物理的にコピーできないようにしてプログラムの不正コピーを防止したり、プログラムにその使用条件を埋め込んで使用時にその使用条件に合致しないときには実行不可能にしてプログラムの不正使用を防止している。

【0004】 ところが、上述のような使用条件でのみプログラムの使用を可能にすると、近年発達してきたローカルエリアネットワーク（LAN）を介して複数のコンピュータを接続して利用するコンピュータ分散処理環境や、コンピュータの設置台数の増加によるコンピュータの使用環境の変化に対応できないという新たな問題が発生した。

【0005】 例えば、ユーザはプログラムの使用権を有するにも係らず、そのプログラムをLAN内の特定のコンピュータのみでしか実行させることができず、その他のコンピュータで実行させたくてもできなかった。また、LAN内のあるコンピュータに使用権が設定されているにも係らず複数のユーザが使用できなくなることがあった。したがって、LANの使用環境を充分に活用することができなかった。

【0006】 そこで、同じプログラムを複数所有しているときにこれらをLAN内の各コンピュータで効率良く使用するために、使用するコンピュータやユーザを特定せずに実行できるようにすると、プログラムが使用権を持たないユーザによって不正に利用されてしまい、プログラムの販売者の利益を守れなくなるという問題があった。

【0007】 従来、フロッピディスク等の記録媒体にソフトウェアのプログラムと共に格納した端末装置のID（マシンID）とソフトウェアIDを比較して、そのソフトウェアの使用ライセンスの正当性をチェックして、ライセンス契約された以外の端末装置で契約対象のソフトウェアが使用されないように防止する装置（例えば、特開平6-119164号公報参照）があった。

## 【0008】

【発明が解決しようとする課題】 上述のような従来の装置では、ユーザはソフトウェアのプログラムと共にマシンIDとソフトウェアIDを格納した記録媒体を持ち回って、LAN上の他の端末装置にソフトウェアIDを移動したり、同じ端末装置への再登録を可能にしているが、使用する端末装置まで記録媒体を持ち回らなければならないので、LAN上の端末装置でソフトウェアのプログラムが不正使用されないようにする保守及び管理が困難になるという問題があった。

【0009】 この発明は上記の点に鑑みてなされたものであり、LAN等の通信手段で接続した複数の端末装置でソフトウェアのプログラムが不正使用されないように

容易に保守及び管理できるようにすることを目的とする。

#### 【0010】

【課題を解決するための手段】この発明は上記の目的を達成するため、サーバと複数の端末装置とを通信手段で接続し、その各端末装置によってプログラムを共用するソフトウェア管理システムにおいて、上記サーバに、上記プログラムの識別名とそのプログラムを使用可能な端末装置の識別名とユーザの識別名とそのユーザのパスワード等の管理情報を記憶する管理情報記憶手段と、その手段に記憶された管理情報と上記端末装置から受信したセキュリティチェック情報とに基づいてその端末装置での上記プログラムの使用可否の妥当性を判断する手段と、その手段による判断結果を上記セキュリティチェック情報を送信してきた端末装置へ送信する手段を設けたものである。

【0011】さらに、上記端末装置に、上記プログラムの実行要求がある度にそのプログラムの識別名と端末装置の識別名とユーザの識別名とそのユーザのパスワード等のセキュリティチェック情報を上記サーバへ送信する手段と、上記サーバから受信した判断結果に基づいて上記プログラムに対する実行権を付与する手段を設けたものである。

【0012】また、上記サーバに、上記端末装置における上記プログラムの実行状態を定期的に問い合わせ、その端末装置から返送されるプログラムの現在の実行状態に基づいてそのプログラムの実行状態を監視する手段を設け、上記端末装置に、上記サーバからの問い合わせがあったときに上記プログラムの現在の実行状態を返送する手段を設けるとよい。

【0013】さらに、上記サーバに、上記端末装置から上記プログラムに対して予め決められた同時実行可能数を越える実行要求を受け付けたとき、及び上記プログラムが予め決められた同時実行可能数で実行されているときに新たな実行要求を受け付けたとき、その実行要求を予め設定した優先度によって調停する手段を設けるとよい。

#### 【0014】

【作用】この発明によるソフトウェア管理システムは、端末装置がプログラムの実行要求がある度にそのプログラムの識別名と端末装置の識別名とユーザの識別名とそのユーザのパスワード等のセキュリティチェック情報をサーバへ送信して、サーバが予め記憶したプログラムの識別名とそのプログラムを使用可能な端末装置の識別名とユーザの識別名とそのユーザのパスワード等の管理情報と端末装置から受信したセキュリティチェック情報とに基づいてその端末装置でのプログラムの使用可否の妥当性を判断して、その判断結果をセキュリティチェック情報を送信してきた端末装置へ送信して、端末装置がサーバから受信した判断結果に基づいてプログラムに対す

る実行権を付与するので、各端末装置でソフトウェアのプログラムが不正使用されないように保守及び管理することが容易に行なえる。

【0015】また、サーバが端末装置におけるプログラムの実行状態を定期的に問い合わせたときに、その端末装置がプログラムの現在の実行状態を返送して、サーバがその端末装置から返送されるプログラムの現在の実行状態に基づいてそのプログラムの実行状態を監視するようにすれば、サーバが各端末装置上でのプログラムの実行状態を常に監視することができ、各端末装置においてプログラムの暴走状態などのようにサーバからの問い合わせに答えられないような状況のときや、プログラムが長時間アイドル状態にあるときには、その端末装置に対するプログラムの実行権（使用権）を停止し、他の端末装置からの実行要求に備えることができ、複数の端末装置でプログラムの使用権を効率良く利用することができる。

【0016】さらに、サーバが、端末装置からプログラムに対して予め決められた同時実行可能数を越える実行要求を受け付けたとき、及びプログラムが予め決められた同時実行可能数で実行されているときに新たな実行要求を受け付けたときには、その実行要求を予め設定した優先度によって調停するようにすれば、サーバはプログラムの実行要求が競合したとき、優先度の高い端末装置に対してプログラムの実行権を許可することができる。また、プログラムの実行要求が競合したとき、緊急度の高い実行要求を終了させずに緊急度の低い実行要求から実行権を委譲させることができる。したがって、ユーザの要求に応じてプログラムの実行権を付与することができる。

#### 【0017】

【実施例】以下、この発明の実施例を図面に基づいて具体的に説明する。図1は、この発明の一実施例であるソフトウェア管理システムの構成を示すブロック図である。このソフトウェア管理システムは、複数の端末装置T1～Tnをお互いに通信手段であるローカルエリアネットワーク（LAN）10で接続し、各端末装置によってプログラムを共用するものである。また、LAN10上で共用されるプログラムのセキュリティチェックサーバとしてサーバSを接続している。

【0018】各端末装置T1～Tnは、パーソナルコンピュータ、ワークステーション、及び特定機能専用端末装置等のコンピュータである。サーバSは、プログラムのセキュリティチェック専用のものでもよいし、パーソナルコンピュータ、ワークステーション等のコンピュータの一プログラムの機能としてセキュリティサーバとしての機能を実行するものでもよい。

【0019】LAN10の伝送形態、接続形態、及びプロトコルは公知のものいずれを使用してもよく、LAN10の延長として考えれば広域ネットワークにも適用

できる。サーバSはこの実施例では1台を接続しているが、複数台を接続して、それぞれをプログラム毎や端末装置毎に用途を使い分けてもよい。

【0020】図2は、図1に示した各端末装置T1～Tnの構成を示すブロック図である。端末装置Tnは、入出力装置1、記憶装置2、LANインタフェース3、及びCPU4からなる。入出力装置1はキーボード、マウス、スキャナ等の入力装置とCRT、LCD等のディスプレイとインクジェットプリンタ及びレーザプリンタ等の出力装置である。

【0021】記憶装置2は、プログラムや各種のデータを格納するハードディスク装置及び光磁気ディスク装置等のメモリである。LANインタフェース3は、LAN10の通信制御を司る。CPU4は、ROM及びRAM等も有するマイクロコンピュータであり、この端末装置全体の制御を司り、この発明に係る機能も果たす。

【0022】すなわち、このCPU4は、プログラムの実行要求がある度にそのプログラムの識別名と端末装置の識別名とユーザの識別名とそのユーザのパスワード等のセキュリティチェック情報をサーバSへ送信する手段と、サーバSから受信した判断結果に基づいてプログラムに対する実行権を付与する手段の機能も果たす。また、サーバSからの問い合わせがあったときにプログラムの現在の実行状態を返送する手段の機能も果たす。

【0023】図3は、図1に示したサーバSの構成を示すブロック図である。サーバSは、入出力装置11、記憶装置12、LANインタフェース13、CPU14、及びタイマ15からなる。入出力装置11はキーボード、マウス、スキャナ等の入力装置とCRT、LCD等のディスプレイとインクジェットプリンタ及びレーザプリンタ等の出力装置である。

【0024】記憶装置12は、プログラムを格納するハードディスク装置及び光磁気ディスク装置等のメモリであり、管理テーブルカウンタ16も有する。この管理テーブルカウンタ16は、管理対象となるプログラムの識別名とそのプログラムを使用可能な端末装置の識別名とユーザの識別名とそのユーザのパスワード等の管理情報を記憶するプログラム管理情報テーブルを格納するエリアである。

【0025】LANインタフェース13は、LAN10の通信制御を司る。タイマ15は、所定時間の割り込み間隔をカウントする。CPU14は、ROM及びRAM等も有するマイクロコンピュータであり、このサーバ全体の制御を司り、この発明に係る機能も果たす。

【0026】すなわち、管理テーブルカウンタ16のプログラム管理情報テーブルに記憶された管理情報と端末装置T1～Tnから受信したセキュリティチェック情報とに基づいてその端末装置でのプログラムの使用可否の妥当性を判断する手段と、その判断結果をセキュリティチェック情報を送信してきた端末装置へ送信する手段の

機能も果たす。

【0027】また、端末装置T1～Tnにおけるプログラムの実行状態を定期的に問い合わせ、その端末装置から返送されるプログラムの現在の実行状態に基づいてそのプログラムの実行状態を監視する手段と、端末装置T1～Tnからプログラムに対して予め決められた同時実行可能数を越える実行要求を受け付けたとき、及びプログラムが予め決められた同時実行可能数で実行されているときに新たな実行要求を受け付けたとき、その実行要求を予め設定した優先度によって調停する手段の機能も果たす。

【0028】このソフトウェア管理システムの初期設定について説明する。まず、サーバSを初期設定する。サーバSの記憶装置12の管理テーブルカウンタ16にプログラム管理情報テーブルを記憶する。

【0029】図6はプログラム管理情報テーブルのフォーマットの一例を示す図である。このプログラム管理情報テーブルは、管理対象となるプログラムのプログラム名(N)毎にプログラムの識別番号(P)と、そのプログラムの同時実行可能数(R)と、そのプログラムを使用可能な端末装置の識別名(Tn)と、各端末装置のプログラム実行優先度(CP)と、プログラム実行中の端末装置からのリクエスト番号と、そのプログラムを使用可能なユーザの識別名(U)と、そのユーザのパスワード(PW)の管理情報を格納している。

【0030】例えば、プログラム名「ワープロB」のプログラムの管理情報は、プログラム識別番号「P05」と同時実行可能数「3」と、使用可能な端末装置の識別名「T1」「T2」「T3」を格納し、その各端末装置のプログラム実行優先度として識別名「T1」の端末装置の優先度「1」と、識別名「T2」の端末装置の優先度「3」と識別名「T3」の端末装置の優先度「2」を格納している。

【0031】また、識別名「T1」「T2」「T3」の各端末装置でプログラム名「ワープロB」のプログラムが実行中のときにはその端末装置から送られて来るリクエスト番号を格納する。このリクエスト番号は実行中ではないときには初期値「0」を格納している。さらに、プログラム名「ワープロB」のプログラムを使用可能なユーザの識別名「U1」「U2」「U3」と、それぞれのユーザのパスワード「PW1」「PW2」「PW3」を格納している。

【0032】この実施例のソフトウェア管理システムでは、このプログラム管理情報テーブル中のプログラム名(N)とプログラム識別番号(P)と同時実行可能数(R)は、そのプログラムの提供者(メーカー等の権利者)がプログラムと一緒にプログラム使用権を得た使用者(例えばサーバの管理者)に提供し、そのプログラムを使用可能な端末装置の識別名(Tn)とユーザ名(U)とユーザのパスワード(PW)はプログラムに依

じてサーバ管理者が設定するものとする。

【0033】さらに、この初期化時にサーバSのタイマ15の所定間隔の割り込み信号発生間隔(TM時間)を設定する。サーバSはこのTM時間毎に管理対象のプログラムを実行中の端末装置に対してそのプログラムの実行状態を問い合わせる。

【0034】次に、このソフトウェア管理システムにおけるプログラムのセキュリティチェック処理について説明する。端末装置は、所望のプログラムの実行を開始するときのプログラム起動時の初期化処理で、サーバSに対して所望のプログラムの識別番号と自装置の識別名とユーザ識別名とそのユーザのパスワードとを格納したログインパケット(loginパケット)をセキュリティチェック情報としてLAN10を介して送り、そのプログラムの実行要求を行なう。

【0035】サーバSはLAN10を介して端末装置からセキュリティチェック情報のログインパケットを受信すると、プログラム管理情報テーブルを参照して、その端末装置におけるプログラム実行(使用)の妥当性を判断する。

【0036】すなわち、サーバSから受け取ったログインパケット内の所望のプログラムの識別番号と自装置の識別名とユーザ識別名とそのユーザのパスワードとが、プログラム管理情報テーブル内の実行要求のあったプログラム名(N)の同時実行可能数(R)と使用可能な端末装置の識別名(Tn)と使用可能なユーザの識別名(U)とそのユーザのパスワード(PW)とに該当するか否かを判断して、該当しなければ「実行不許可」を、該当すれば「実行許可」を、その判断結果として端末装置へ返送する。

【0037】端末装置は、サーバSから実行要求したプログラムに対する実行の妥当性の判断結果を受け取ると、それに基づいてプログラムの動作を開始してもよい。か否かを判断し、サーバSから受け取った判断結果が「実行許可」ならプログラムの実行を開始し、「実行不許可」ならプログラムの初期化処理を終了して実行を行なわない。なお、サーバSから「実行不許可」を受け取ったときにプログラムの実行を行なわないときの処理はプログラム毎に任せても良い。

【0038】サーバSは、各端末装置でプログラム実行中はタイマ15がTM時間毎に定期的に発生する割り込み信号をトリガにして、プログラム管理情報テーブルをサーチし、プログラム実行中の端末装置を検索し、検索された端末装置に対してリクエスト番号を付けて、順に問い合わせのパケット(Q)であるチェックパケット(checkパケット)を送信する。

【0039】端末装置は、サーバSからチェックパケットを受け取ると、問い合わせのあったプログラムが実行中であれば回答パケット(A)を返送する。サーバSは、チェックパケットの送信先の端末装置から所定時間

(TMX時間)内に回答のパケットを受け取らなかったときには、そのプログラムが端末装置で暴走状態や異常終了等の原因で応答がないものと判断して、その端末装置におけるプログラムのリクエスト番号を初期値「0」にし、実行中を削除する。このTMX時間は数十秒から数分にするとよい(例えば60秒)。

【0040】そして、端末装置は、プログラム終了時には終了処理の中でサーバSにログアウト(logout)の問い合わせのパケット(Q)を送信して、サーバSはそのパケットを受信すると、そのパケット中の該当する端末装置のプログラムのリクエスト番号を初期値「0」にして、実行中を削除する。このようにして、一連の処理を完了する。

【0041】図4は、端末装置とサーバとの間でやり取りされる問い合わせのパケットのフォーマットの一例を示す図である。このパケットには、ヘッダとして問い合わせのパケットの識別情報(Q)を格納している。

【0042】端末装置がこのパケットをプログラム実行要求のセキュリティチェックをサーバに対して要求するときのログインパケットとして使用する場合、プログラムの実行要求に対するセキュリティチェックの問い合わせとして、端末装置の識別名としてコンピュータ番号と、リクエスト番号である識別番号とプログラムの識別名である識別番号とユーザの識別名であるユーザ名とそのユーザのパスワードを格納する。

【0043】リクエスト識別番号は、端末装置が最初のログイン時に付ける「0」以外のユニークな番号であり、サーバSから見たときに端末装置上のプログラムを識別するために使用する番号であり、ログアウト(logout)まで有効である。

【0044】このリクエスト識別番号は、図6に示した使用可能な端末装置の識別名(Tn)に対応させており、実行中でないときには初期値「0」を格納し、実行中であるときにはその端末装置から送られて来る数値を格納する。このリクエスト番号をサーチすることによって初期値「0」でない端末装置ではプログラムが実行中であることを判断することができる。

【0045】また、サーバSがこのパケットをプログラム実行中の端末装置に対する問い合わせのチェックパケットとして使用する場合、プログラム実行中の問い合わせとして、問い合わせ先の端末装置のコンピュータ番号と、リクエスト番号などを付けて端末装置へ送信する。

【0046】図5は、端末装置とサーバとの間でやり取りされる回答のパケットのフォーマットの一例を示す図である。このパケットにはヘッダとして回答のパケットの識別情報(A)を格納している。このパケットは、端末装置からのプログラム実行要求のセキュリティチェックの問い合わせに対する回答のパケット、及び端末装置がこのパケットをプログラム実行中のサーバからの問い合わせに対する回答のパケットとして使用する。

【0047】サーバSがこのパケットを端末装置からのプログラム実行要求のセキュリティチェックの問い合わせに対する回答のパケットとして使用する場合、プログラム実行の可否を伝えるパケットとして、プログラムのセキュリティチェックの問い合わせをした端末装置のコンピュータ番号と、その端末装置からのリクエスト番号と、実行要求のあったプログラムの識別番号と、そのプログラムの実行の許可又は不許可を伝える情報と、不許可のときにはその不許可理由を示す情報を格納する。

【0048】図7は、図5のパケットに格納するプログラムの実行の許可と不許可を伝える情報のコードの一例を示す図である。サーバSは、セキュリティチェックの結果により、上記パケットにプログラムの実行を許可するときには「OK；実行許可」を示すコード「0」を格納し、実行が不許可のときには「NG；実行不許可」を示すコード「1」を格納して端末装置へ返送する。

【0049】図8は、図5のパケットに格納するプログラム実行の不許可理由を伝える情報のコードの一例を示す図である。サーバSは、セキュリティチェックの結果により、プログラムの実行を不許可するときには、上記の不許可を示すコード「1」と共に、その不許可理由を示すコードも格納して端末装置へ返送する。

【0050】その不許可理由として、「プログラム識別番号未登録」の場合はコード「1」を、コンピュータ名（端末装置の識別名）未登録の場合はコード「2」を、同時実行可能数を越えている場合はコード「3」を、未定義ユーザ（使用可能なユーザ以外のユーザ）の場合はコード「4」を、パスワードが違う場合はコード「5」を予期しないエラーの場合はコード「9」を格納する。なお、不許可理由はその他にもあるがここでは省略する。

【0051】図9は、セキュリティチェック時のサーバと端末装置との通信手順を示す図である。まず、端末装置Tnはプログラム起動時の初期化処理においてサーバSへログインパケット（Q）を送信する。

【0052】サーバSはそのログインパケット（Q）を受信すると、その端末装置Tnから実行要求のあったプログラムのセキュリティチェックを行ない、実行許可と判断したらそれを示す回答パケット（A）を返送する。端末装置は実行許可の回答パケットを受け取るとそのプログラムの実行を開始する。

【0053】サーバSは、端末装置からログインパケット（Q）を受け取るとタイマ15によるTM時間毎の割り込み信号によって、管理対象に有るプログラムが実行中の端末装置をサーチし、サーチされた端末装置Tnに対してプログラムの実行状況を問い合わせるチェックパケット（Q）を送信する。そして、端末装置Tnにおいてプログラムが異常なく実行中ならば回答パケット

（A）が返送されてくるので、その場合にはチェックを終了する。

【0054】また、チェックパケット（Q）を送信してからTMX時間以内に回答パケット（A）が返送されてこないときには、その端末装置におけるプログラムの実行中を示すリクエスト番号を初期化する。そして、端末装置はプログラムを終了したらサーバSへログアウトパケット（Q）を送信し、サーバSはそのログアウトパケット（Q）を受信するとその端末装置におけるプログラムの実行中を示すリクエスト番号を初期化する。

【0055】この実施例のソフトウェア管理システムでは、サーバを用いて各端末装置でソフトウェアのプログラムが不正使用されないように保守及び管理することが容易に行なえる。また、サーバが各端末装置のプログラムの実行状況を監視するので、プログラムの暴走状態や異常終了などのときには、その端末装置に対するプログラムの実行権を停止し、他の端末装置からの実行要求に備えることができるので、複数の端末装置でプログラムの使用権を効率良く利用することができる。

【0056】次に、この実施例のソフトウェア管理システムのサーバによって各端末装置におけるプログラムのアイドル状態を監視する場合の処理について説明する。この場合、端末装置Tnの記憶装置内に備えたカウンタによってカウントしたプログラムの実行時間を用いる。この実行時間はプログラムが実際に使用したCPU時間であり、例えば、UNIXのOSの場合は「プログラムがユーザ空間で使用したCPU時間」とする。

【0057】サーバSは、タイマ15によってTM時間毎に定期的に発生される割り込み信号をトリガにして、TM時間毎にプログラム実行中の端末装置Tnへプログラムの実行状況を問い合わせるチェックパケットを送信する。端末装置Tnはそのチェックパケットを受信すると、該当するプログラムが実行中であればカウンタの内容を回答パケット（A）に格納してサーバSへ返送する。

【0058】サーバSは端末装置から送信される回答パケット（A）に格納されたカウンタの内容に基づいて、その端末装置のプログラムの実行時間の変化を監視し、変化がないときには、プログラムが実行中であるにも係らず実際にはアイドル状態が続いていると判断して、そのプログラムの終了を要求するパケットであるキルパケット（killパケット）を端末装置Tnへ送信する。端末装置Tnではキルパケットを受信すると、サーバSに対して肯定応答パケット（A）を返送すると共に、アイドル状態のプログラムを終了させる。

【0059】図10は、セキュリティチェック時のサーバと端末装置との他の通信手順を示す図である。まず、端末装置Tnはプログラム起動時の初期化処理においてサーバSへログインパケット（Q）を送信する。

【0060】サーバSはそのログインパケット（Q）を受信すると、その端末装置Tnから実行要求のあったプログラムのセキュリティチェックを行ない、実行許可と

判断したらそれを示す回答パケット (A) を返送する。端末装置は実行許可の回答パケットを受け取るとそのプログラムの実行を開始する。

【0061】サーバSは、端末装置からログインパケット (Q) を受け取るとタイマ15によるTM時間毎の割り込み信号によって、管理対象に有るプログラムが実行中の端末装置をサーチし、サーチされた端末装置Tnに対してプログラムの実行状況を問い合わせるチェックパケット (Q) を送信する。

【0062】端末装置Tnはそのチェックパケット (Q) を受け取ると、カウンタの内容を格納した回答パケット (A) をサーバSへ返送する。サーバSはチェックパケット (Q) を送信してからTMX時間以内に回答パケット (A) が返送されてきたら、その回答パケット (A) に格納されているカウンタの内容であるCPU実行時間を前回に返送されてきたCPU実行時間と比較する。

【0063】通常実行時には、TMX時間以内に返送された回答パケット (A) に格納されたCPU実行時間は、前回にチェックしたときのCPU実行時間から変化しているので、プログラムの実行の継続を容認する。しかし、前回にチェックしたときのCPU時間を前回と比較して変化していなければアイドル状態と判断して、その端末装置へプログラム終了要求のキルパケット (Q) を送信する。

【0064】端末装置は、そのキルパケット (Q) を受信すると、肯定応答パケット (A) を返送し、そのプログラムを終了させる。サーバSは、端末装置から肯定応答パケット (A) を受け取ると、その端末装置のプログラムの実行を示すリクエスト番号を初期化する。

【0065】このようにして、サーバが各端末装置上でのプログラムの実行状態を常に監視することができ、各端末装置においてプログラムが長時間アイドル状態にあるときには、その端末装置に対するプログラムの実行を停止させて、他の端末装置からの実行要求に備えることができ、複数の端末装置でプログラムの使用権を効率良く利用することができる。

【0066】次に、この実施例のソフトウェア管理システムのサーバによる各端末装置に対するプログラムの実行権の調停処理について説明する。サーバSは、端末装置Tnからプログラムに対して予め決められた同時実行可能数を越える実行要求を受け付けたとき、プログラムが予め決められた同時実行可能数で実行されているときに新たな実行要求を受け付けたときには、その実行要求を予め設定した優先度によって調停する。

【0067】すなわち、サーバSは、端末装置からログインパケットを受け付けたとき、そのパケットによる実行要求があったプログラムについてのセキュリティチェックを行ない、使用可能な端末装置とユーザであり、同時実行可能数にまだ余裕があるときには、すぐにその端

末装置にプログラム実行の許可を与える。

【0068】また、同時実行可能数に余裕がないときには、プログラム管理情報テーブルのプログラム実行優先度を参照して、その端末装置よりも優先度の低い端末装置でプログラム実行中のものをサーチし、そのサーチした端末装置に対するプログラム実行をキル (kill) し、それに代えて優先度の高い端末装置に実行権を許可する。

【0069】図11は、セキュリティチェック時のサーバと端末装置とのさらに他の通信手順を示す図である。ここでは、図6に示したプログラム管理情報テーブルのプログラム名「表計算C」を使用する端末装置T1、T2、T3、T4、T5のプログラム実行権を調停する場合について説明する。このプログラム名「表計算C」の同時実行可能数は「4」である。

【0070】まず、サーバSは端末装置T1、T2、T4に対して「表計算C」のプログラムの実行を許可し、端末装置T1、T2、T4ではそれぞれそのプログラムを実行中である。

【0071】サーバSは、端末装置T5からプログラム識別名「P06」の「表計算C」のプログラムの実行要求のログインパケット (Q) を受信すると、使用可能な端末装置と使用可能なユーザであることを確認し、同時実行可能数に余裕があるため、端末装置T5にプログラム「表計算C」の実行権を許可する回答パケット (A) を送る。端末装置T5ではその回答パケット (A) を受信すると共に、プログラム「表計算C」の実行を開始する。

【0072】サーバSは、各端末装置T1、T2、T4、T5に対してそれぞれのログインパケット (Q) を受信してからTM時間毎にプログラムの実行状況を問い合わせて監視する。

【0073】その際、TMX時間以内に応答がないときにはプログラムの暴走あるいは異常終了と判断してプログラム管理情報テーブルのリクエスト番号を初期化して、プログラム実行可能数を確保する。また、各端末装置からの回答パケット (A) に格納されたCPU時間を前回と比較して変化があればそのまま継続させるが、変化がなければアイドル状態と判断して、その端末装置に対してプログラムを終了させて、プログラム実行可能数を確保する。

【0074】その後、サーバSは、端末装置T3からプログラム識別名「P06」の「表計算C」のプログラムの実行要求のログインパケット (Q) を受信すると、使用可能な端末装置と使用可能なユーザであることを確認した後、同時実行可能数に余裕がないため、プログラム実行優先度を参照して、端末装置T3よりも低い優先度を持ち現在プログラム「表計算C」を実行中の端末装置をサーチする。

【0075】この場合、端末装置T5がそれに該当する



ので、サーバSは端末装置T5に対してキルパケット(Q)を送信して、プログラム「表計算C」の稼働の停止要求を行ない、端末装置T5はそのキルパケット(Q)に基づいてプログラム「表計算C」の稼働を終了して実行権を剥奪し、サーバSへ肯定回答(A)を返送して実行権を返上する。

【0076】サーバSは端末装置T5から肯定回答

(A)を受信すると、プログラム管理情報テーブルの端末装置T5のリクエスト番号を初期化し、端末装置T3から受け取ったリクエスト番号を格納して、端末装置T3に実行許可の回答パケット(A)を送信する。端末装置T3は、サーバSから回答パケット(A)を受信すると、プログラム「表計算C」の実行権を付与し、そのプログラムを実行させる。

【0077】また、例えば、各端末装置T1～T5から同じプログラムの実行要求があったときにも、各端末装置T1～T5の優先度順に同時実行可能数まで実行許可を通知し、同時実行可能数を越えてしまう端末装置に対しては実行不許可を通知して、複数の端末装置の実行要求の競合の調停を行なうこともできる。

【0078】なお、上述の実施例では、プログラムは各端末装置に予め格納されており、サーバSからの許可通知があったときのみ実行する場合について説明したが、サーバSに各端末装置が共用するプログラムを格納し、各端末装置からの実行要求があり、その実行を許可すると共にプログラムの利用を可能にするようにしてもよい。

【0079】また、ワープロや表計算等のメインプログラムに初期化処理のプログラムもセットされているプログラムでは、その初期化処理時にメインプログラムの実行要求があったときに、上述の処理を行なうようにすればよい。

【0080】さらに、プログラムの識別名又は識別番号、プログラムの同時実行可能数、端末装置名又は端末番号、ユーザ識別名又はユーザ識別番号、ユーザのみが知りうるパスワードを用いて、プログラムのセキュリティチェックでプログラムの実行の妥当性を判断する項目としており、プログラム毎に予め決め手おいた条件と一致する場合にのみ実行許可を与えているので、プログラムの不正使用をより効果的に防止することができる。

【0081】その識別名と識別番号をネットワーク上でユニークなものにすれば、複数のネットワークを接続したシステムに適用する場合、ネットワーク毎にプログラムのセキュリティチェックでプログラムの実行の妥当性を判断する項目を異ならせることができ、誤って異なるシステムの端末装置にプログラム実行の許可を与えてしまう恐れがなく、広域ネットワークにおけるプログラムの不正使用を防止することができる。

【0082】このようにして、サーバが、予め決められた同時実行可能数を越える実行要求を受け付けたとき、

予め決められた同時実行可能数が実行されているときに新たな実行要求を受け付けたとき、及び同時実行可能数を越える端末装置からのプログラムの実行要求が競合したとき、優先度の高い端末装置に対して優先的にプログラムの実行権を許可することができるので、緊急度の高い実行要求を終了させずに済む。したがって、ユーザの要求に応じてプログラムの実行権を調停することができ、複数の端末装置でプログラムの使用権を効率良く利用することができる。

【0083】この実施例のソフトウェア管理システムは、プログラムの不正コピー及び不正使用を防止し、且つコンピュータネットワークシステム上の各端末装置でプログラムの実行に支障がないようにすることができる。

【0084】また、サーバが各端末装置上でのプログラムの実行状態を常に監視するため、端末装置上でプログラムが暴走状態や異常終了したときに実行権をサーバに取り戻して、他の端末装置からの実行要求に備えることができ、端末装置上でプログラムが長時間アイドル状態にあるときにも、実行権をサーバに取り戻して、他の端末装置からの実行要求に備えることができる。

【0085】さらに、サーバが各端末装置に対するプログラムの実行権を各端末装置に設定された優先度に基づいて調停するので、ユーザに応じたプログラム実行権の付与を行なえる。したがって、ネットワーク上でのプログラムの不正使用を防止し、ネットワーク上の各端末装置を有効に使用してプログラムの使用権を効率良く利用することができる。

【0086】

【発明の効果】以上説明してきたように、この発明によるソフトウェア管理システムによれば、LAN等の通信手段で接続した複数の端末装置でソフトウェアのプログラムが不正使用されないように保守及び管理を容易に行なえる。

【図面の簡単な説明】

【図1】この発明の一実施例であるソフトウェア管理システムの構成を示すブロック図である。

【図2】図1に示した各端末装置T1～Tnの構成を示すブロック図である。

【図3】図1に示したサーバSの構成を示すブロック図である。

【図4】図1に示した端末装置とサーバとの間でやり取りされる問い合わせのパケットのフォーマットの一例を示す図である。

【図5】図1に示した端末装置とサーバとの間でやり取りされる回答のパケットのフォーマットの一例を示す図である。

【図6】図3に示したサーバSの記憶装置12に格納するプログラム管理情報テーブルのフォーマットの一例を示す図である。



【図7】図5のパケットに格納するプログラムの実行の許可と不許可を伝える情報のコードの一例を示す図である。

【図8】図5のパケットに格納するプログラム実行の不許可理由を伝える情報のコードの一例を示す図である。

【図9】図1に示したソフトウェア管理システムにおけるセキュリティチェック時のサーバと端末装置との通信手順を示す図である。

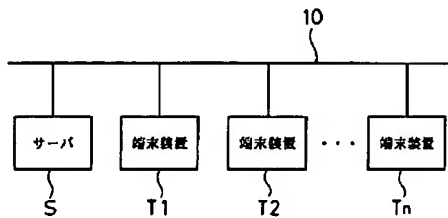
【図10】図1に示したソフトウェア管理システムにおけるセキュリティチェック時のサーバと端末装置との他の通信手順を示す図である。

【図11】図1に示したソフトウェア管理システムにおけるセキュリティチェック時のサーバと端末装置とのさらに他の通信手順を示す図である。

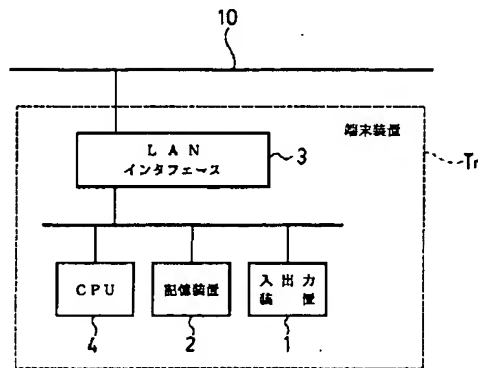
【符号の説明】

1, 11：入出力装置      2, 12：記憶装置  
3, 13：LANインタフェース  
4, 14：CPU      15：タイマ  
16：管理テーブルカウンタ  
10：ローカルエリアネットワーク（LAN）  
T1～Tn：端末装置      S：サーバ

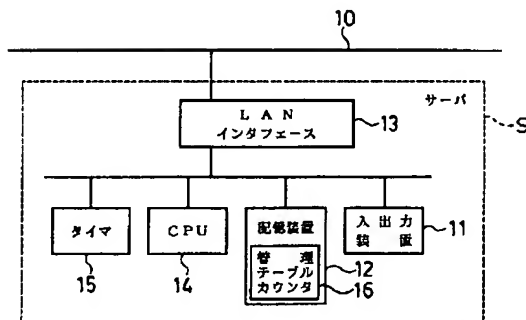
【図1】



【図2】



【図3】



【図4】

| 問い合わせ     | 型   |
|-----------|-----|
| "Q"       | 文字  |
| コンピュータ番号  | 数値  |
| リクエスト識別番号 | 数値  |
| プログラム識別番号 | 数値  |
| ユーザ名      | 文字列 |
| パスワード     | 文字列 |

【図5】

| 回答        | 型   |
|-----------|-----|
| "A"       | 文字  |
| コンピュータ番号  | 数値  |
| リクエスト識別番号 | 数値  |
| プログラム識別番号 | 数値  |
| 許可/不許可    | 文字列 |
| 不許可理由     | 文字列 |

【図7】

| 値 | 意味        |
|---|-----------|
| 0 | OK: 実行許可  |
| 1 | NG: 実行不許可 |

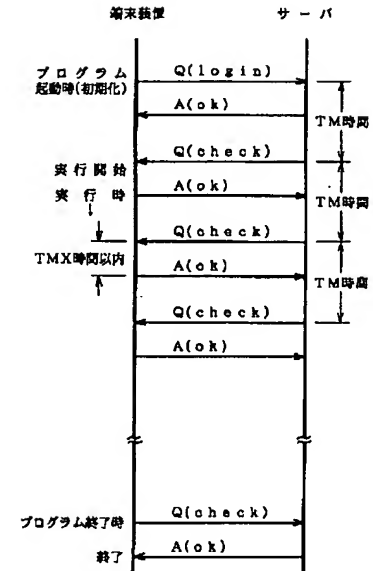
【図8】

| 値   | 意味           |
|-----|--------------|
| 1   | プログラム識別番号未登録 |
| 2   | コンピュータ名未登録   |
| 3   | 同時使用数を超えた    |
| 4   | 未定義ユーザ       |
| 5   | パスワードが違う     |
| ... | ...          |
| 9   | 予期しないエラー     |

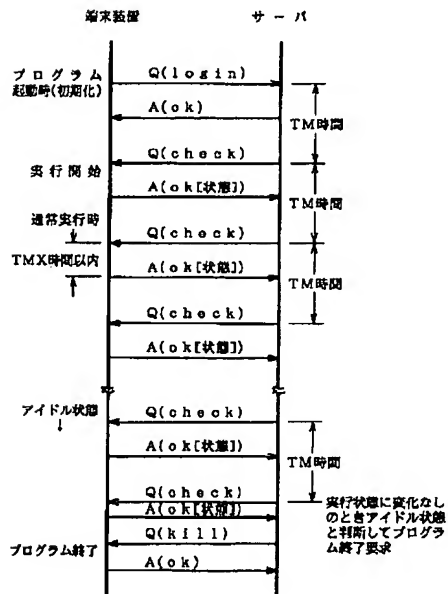
【図6】

| 番号  | プログラム名<br>(N) | プログラム<br>識別番号<br>(P) | 同時実行<br>可能数<br>(R) | 使用可能な<br>端末数<br>(Ta)       | プログラム<br>実行優先度<br>(CP) | リクエスト<br>番号<br>(初期値0) | 使用可能な<br>ユーザ<br>(U)  | ユーザの<br>パスワード<br>(PW)    |
|-----|---------------|----------------------|--------------------|----------------------------|------------------------|-----------------------|----------------------|--------------------------|
| 型   | 文字列           | 数値                   | 数値                 | 数値                         | 数値                     | 数値                    | 文字列                  | 文字列                      |
| 1   | ワープロA         | P01                  | 2                  | T1<br>T3                   | 1<br>2                 |                       | U1<br>U2             | PW1<br>PW2               |
| 2   | ワープロB         | P05                  | 3                  | T1<br>T2<br>T3             | 1<br>3<br>2            |                       | U1<br>U2<br>U3       | PW1<br>PW2<br>PW3        |
| 3   | 表計算C          | P08                  | 4                  | T1<br>T2<br>T3<br>T4<br>T5 | 6<br>4<br>3<br>2<br>1  |                       | U1<br>U2<br>U3<br>U4 | PW1<br>PW2<br>PW3<br>PW4 |
| ... | ...           | ...                  | ...                | ...                        | ...                    | ...                   | ...                  | ...                      |
| m   | 表計算D          | P10                  | 1                  | T5                         | 1                      |                       | U1<br>U5             | PW1<br>PW5               |

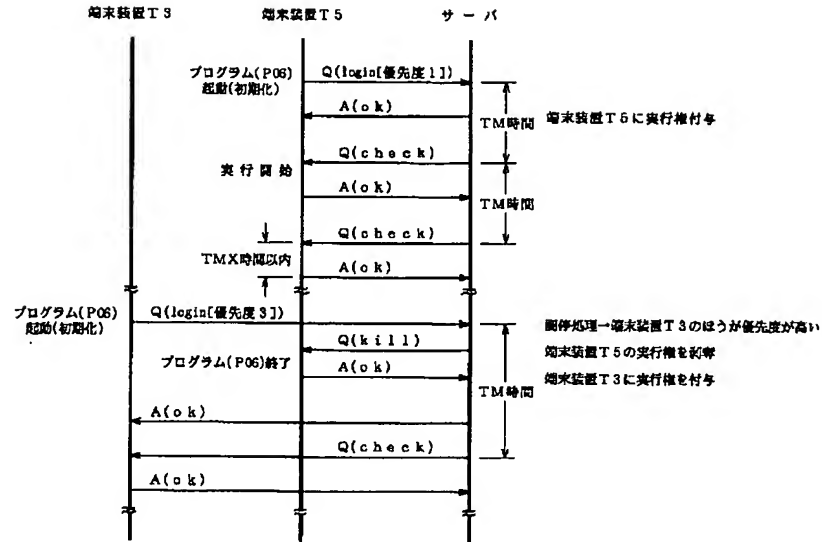
【図9】



【図10】



【図11】



# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-263283

(43)Date of publication of application : 11.10.1996

---

(51)Int.Cl. G06F 9/06  
G06F 15/00

---

(21)Application number : 07-066081 (71)Applicant : RICOH CO LTD

(22)Date of filing : 24.03.1995 (72)Inventor : KOBAYASHI HIDEKI

---

## (54) SOFTWARE MANAGING SYSTEM

### (57)Abstract:

PURPOSE: To easily maintain and manage the program of software so as not to be illegally used in plural terminal equipments connected by a communication means such as LAN.

CONSTITUTION: Each time of the execution request of the program terminal equipment T<sub>n</sub> transmits security check information such as the identification names of the program, the terminal equipment and a user to a server S. Then the server S judges the validity of the using possibility of the program in the terminal equipment T<sub>n</sub> based on managing information such as the identification name of the program, the identification name of the terminal equipment capable of using it and the identification name of the user which are stored in advance and security check information received from the terminal equipment T<sub>n</sub>. Then the judging result is transmitted to the terminal equipment T<sub>n</sub> which transmitted security information and then the terminal equipment T<sub>n</sub> gives an execution right to the program based on the judging result received from the server S.

---

## CLAIMS

---

[Claim(s)]

[Claim 1] A software management system which connects a server and two or more terminal units by a means of communication and shares a program with this each terminal unit comprising:

A management information storage means which memorizes management informations such as an identifier of an usable terminal unit, a user's identifier, and this user's password for an identifier of said program and this program to said server.

A means to judge the validity of use propriety of said program in this terminal unit based on management information memorized by this means and security check information received from said terminal unit.

A means to transmit a decision result by this means to a terminal unit which has transmitted said security check information is formed. A means to transmit security check information including an identifier of this program, an identifier of a terminal unit, a user's identifier, this user's password, etc. to said server whenever said terminal unit has an execution demand of said program.

A means to give an execution right over said program based on a decision result received from said server.

[Claim 2] A software management system comprising:

A means to supervise a run state of the program based on the present run state of a program which asks said server periodically a run state of said program in said terminal unit and is returned to it from this terminal unit in the software management system according to claim 1.

A means to return the present run state of said program when said terminal unit has an inquiry from said server.

[Claim 3]When an execution demand which exceeds a concurrency possible number beforehand decided to be said server from said terminal unit to said program in the software management system according to claim 1 is receivedAnd a software management system forming a means to arbitrate this execution demand with a priority set up beforehand when an execution demand new when said program is executed with a concurrency possible number decided beforehand is received.

---

## DETAILED DESCRIPTION

---

[Detailed Description of the Invention]

[0001]

[Industrial Application]This invention connects a server and two or more terminal units by the means of communication of a Local Area Network (LAN) etc.and relates to the software management system which shares various kinds of programs with each of that terminal unit.

[0002]

[Description of the Prior Art]When using the program of a computer nowacquiring and using the right (a "license" is called) for a user to exchange a contract among the venders of a program and to use the program is called for. And service conditions which use it for a programsuch as a computera userand the number of concurrent usewere imposedand except the service conditionas use of the program was not acceptedthe unauthorized use of a program is prevented.

[0003]Permission is made for there to be nothingas a program can be copied physicallywhen not preventing the illegal copy of a programor embedding the service condition at a program and not agreeing in the service condition at the time of useexecution is made impossible and the unauthorized use of a program is prevented.

[0004]Howeverif use of a program is enabled only on the above service

conditionsThe new problem that it could not respond to change of the computer distributed processing environment which connects and uses two or more computers via the Local Area Network (LAN) which has progressed in recent yearsand the operating environment of the computer by the increase in the number of install stands of a computer occurred.

[0005]For examplealthough the user had a royalty of a programeven if the program is not performed only by specific computer in LAN and he wanted to make it perform by other computershe was not able to do it. There was a thing for which the royalty is set as a certain computer in LAN and it becomes impossible to use two or more users in spite of being needed. Thereforethe operating environment of LAN was not fully able to be utilized.

[0006]Thenwhen two or more same programs are ownedin order to use these efficiently by each computer in LANWhen it enabled it to perform without specifying the computer and user who use ita program will be unjustly used by the user without a royalty and there was a problem of it becoming impossible to protect the profits of the vender of a program.

[0007]ID (machine ID) and software ID of a terminal unit which were stored in recording mediasuch as a floppy diskwith the program of software are compared conventionallyThere was a device (for examplerefer to JP6-119164A) prevented so that the justification of the license to use of the software may be checked and the software for a contract may not be used with the terminal unit except the licensing agreement having been carried out.

[0008]

[Problem(s) to be Solved by the Invention]Although a user turns with the recording medium which stored machine ID and software ID with the program of softwaresoftware ID is moved to other terminal units on LAN or re-registration to the same terminal unit is enabled in the above conventional devicesSince even the terminal unit to be used had to turn with the recording mediumthere was a problem that the maintenance and management which are kept from the program of software being used improperly with the terminal unit on LAN became difficult.



[0009]This invention is made in view of the above-mentioned pointand it aims at enabling it to perform maintenance and manage easily so that the program of software may not be used improperly with two or more terminal units connected by means of communicationsuch as LAN.

[0010]

[Means for Solving the Problem]In a software management system which connects a server and two or more terminal units by a means of communicationand shares a program with each of that terminal unit in order that this invention may attain the above-mentioned purposeA management information storage means which memorizes management informationsuch as a user's identifier of an usable terminal unitan identifiera user's passwordetc.for an identifier and a program of the above-mentioned program to the above-mentioned serverA means to judge the validity of use propriety of the above-mentioned program in the terminal unit based on management information memorized by the means and security check information received from the above-mentioned terminal unitA means to transmit a decision result by the means to a terminal unit which has transmitted the above-mentioned security check information is formed.

[0011]A means to transmit security check informationincluding a user's identifier of the programan identifier of a terminal unitan identifiera user's passwordetc.to the above-mentioned server whenever the above-mentioned terminal unit has an execution demand of the above-mentioned programA means to give an execution right over the above-mentioned program based on a decision result received from the above-mentioned server is formed.

[0012]A run state of the above-mentioned program in the above-mentioned terminal unit is periodically asked to the above-mentioned serverWhen a means to supervise a run state of the program based on the present run state of a program returned from the terminal unit is formed and the above-mentioned terminal unit has an inquiry from the above-mentioned serverit is good to form a means to return the present run state of the above-mentioned program.

[0013]When an execution demand exceeding a concurrency possible number beforehand decided to be the above-mentioned server from the above-mentioned terminal unit to the above-mentioned program is receivedAnd when an execution demand new when the above-mentioned program is executed with a concurrency possible number decided beforehand is receivedin additionit is good to form a means to arbitrate the execution demand with a priority set up beforehand.

[0014]

[Function]The software management system by this inventionWhenever an execution demand of a program has a terminal unitsecurity check informationincluding a user's identifier of the programan identifier of a terminal unitan identifiera user's passwordetc.is transmitted to a serverThe identifier and program of the program which the server memorized beforehand. Based on the identifier of an usable terminal unita user's identifiermanagement informationsuch as the user's passwordand the security check information received from the terminal unitthe validity of the use propriety of the program in the terminal unit is judgedSince the decision result is transmitted to the terminal unit which has transmitted security check information and a terminal unit gives the execution right over a program based on the decision result received from the serverIt can perform easily performing maintenance and managing so that the program of software may not be used improperly with each terminal unit.

[0015]When a server asks the run state of the program in a terminal unit periodicallyIf the terminal unit returns the present run state of a program and a server supervises the run state of the program based on the present run state of the program returned from the terminal unitA server can always supervise the run state of the program on each terminal unitThe time of the situation which cannot reply to the inquiry from a server like the runaway state of a program in each terminal unitand when a program is in an idle state for a long timeThe execution right (royalty) of the program over the terminal unit can be stoppedit can prepare for the execution demand from other terminal unitsand the royalty of a program

can be efficiently used with two or more terminal units.

[0016]When the execution demand for which a server exceeds the concurrency possible number beforehand decided from the terminal unit to the program is receivedAnd when an execution demand new when the program is executed with the concurrency possible number decided beforehand is receivedIf the execution demand is arbitrated with the priority set up beforehandthe server can permit the execution right of a program to a terminal unit with a high prioritywhen an execution demand of a program competes. When an execution demand of a program competesan execution right can be made to transfer from a low execution demand of urgencywithout terminating an urgent execution demand. Thereforethe execution right of a program can be given according to a user's demand.

[0017]

[Example]Hereafterthe example of this invention is concretely described based on a drawing. Drawing 1 is a block diagram showing the composition of the software management system which is one example of this invention. This software management system connects two or more terminal units T1-Tn in Local Area Network (LAN) 10 which is a means of communication mutuallyand shares a program with each terminal unit. The server S is connected as a security check server of the program shared on LAN10.

[0018]Each terminal units T1-Tn are computerssuch as a personal computera workstationand a specific function dedicated terminal. The thing only for a security check of a program may be sufficient as the server Sand it may perform the function as a security server as a function of one program of computerssuch as a personal computer and a workstation.

[0019]The transmission forms of LAN10a topologyand a protocol may use any of a publicly known thingand if it thinks as extension of LAN10they can apply them also to a wide area network. Although the server S has connected one set in this exampleit may connect two or more sets and may use a use for each properly for every program and every terminal unit.

[0020]Drawing 2 is a block diagram showing the composition of each terminal units T1-Tn shown in drawing 1. The terminal unit Tn consists of the input/output device 1, the memory storage 2, LAN interface 3, and CPU 4. The input/output devices 1 are output units such as input devices such as a keyboard, a mouse, and a scanner, a display of CRT, LCD, etc., an ink-jet printer, and a laser beam printer.

[0021]The memory storage 2 is memories which store a program and various kinds of data such as a hard disk drive and optical-magnetic disc equipment. LAN interface 3 manages the communications control of LAN 10. CPU 4 is a microcomputer which has ROM, RAM, etc.

Control of this whole terminal unit is managed and the function concerning this invention is also achieved.

[0022]Namely, a means to transmit security check information including a user's identifier of that program, an identifier of a terminal unit, an identifier of a user's password, etc., to the server S whenever this CPU 4 has an execution demand of a program. The function of a means to give the execution right over a program based on the decision result received from the server S is achieved. When there is an inquiry from the server S, the function of a means to return the present run state of a program is also achieved.

[0023]Drawing 3 is a block diagram showing the composition of the server S shown in drawing 1. The server S consists of the input/output device 11, the memory storage 12, LAN interface 13, CPU 14, and the timer 15. The input/output devices 11 are output units such as input devices such as a keyboard, a mouse, and a scanner, a display of CRT, LCD, etc., an ink-jet printer, and a laser beam printer.

[0024]The memory storage 12 is memories which store a program such as a hard disk drive and optical-magnetic disc equipment.

It also has the management table counter 16.

This management table counter 16 is an area which stores the program-management-information table which memorizes management information such as a user's identifier of an usable terminal unit, an identifier of a user's

password etc. for the identifier and program of the program used as an administration object.

[0025] LAN interface 13 manages the communications control of LAN10. The timer 15 counts the interruption interval of predetermined time. CPU14 is a microcomputer which has ROM/ RAM etc.

Control of this whole server is managed and the function concerning this invention is also achieved.

[0026] Namely, it means to judge the validity of the use propriety of the program in the terminal unit based on the management information memorized on the program-management-information table of the management table counter 16 and the security check information received from the terminal units T1-Tn. The function of a means to transmit the decision result to the terminal unit which has transmitted security check information is achieved.

[0027] A means to supervise the run state of the program based on the present run state of the program which asks periodically the run state of the program in the terminal units T1-Tn and is returned from the terminal unit. When the execution demand exceeding the concurrency possible number beforehand decided from the terminal units T1-Tn to the program is received, and when an execution demand new when the program is executed with the concurrency possible number decided beforehand is received, the function of a means to arbitrate with the priority which set up the execution demand beforehand is also achieved.

[0028] Initial setting of this software management system is explained. First, the server S is initialized. A program-management-information table is memorized at the management table counter 16 of the memory storage 12 of the server S.

[0029] Drawing 6 is a figure showing an example of a format of a program-management-information table. The program name of the program from which this program-management-information table serves as an administration object -- (with the identification number (P) of a program to every N). The concurrency possible number (R) and program of the program The identifier (Tn) of an usable

terminal unitThe management information of the identifier (U) and a user's password of an usable user (PW) is stored for the program execution priority (CP) of each terminal unitrequest numberand program from the terminal unit under program execution.

[0030]For examplethe management information of the program of a program name "word processor B"Store a program identification number "P05"a concurrency possible number "3"the identifier "T1" of an usable terminal unitT2and "T3"and as a program execution priority of each of that terminal unit The priority "1" of the terminal unit of an identifier "T1"The priority "3" of the terminal unit of an identifier "T2" and the priority "2" of the terminal unit of an identifier "T3" are stored.

[0031]While the program of a program name "word processor B" is performing with each terminal unit of an identifier "T1"T2and "T3"the request number sent from the terminal unit is stored. This request number stores the initial value "0"when it is not under execution. An usable user's identifier "U1"U2U3and the password "PW1"PW2and "PW3" of each user are stored for the program of a program name "word processor B."

[0032]In the software management system of this example. The program name (N)program identification number (P)and concurrency possible number (R) in this program-management-information tableThe donor (right holderssuch as a maker) of the program provides for the user (for exampleadministrator of a server) who acquired the program royalty together with the programAccording to a programas for the password (PW) of the identifier (Tn) of an usable terminal unita user name (U)and a usera server manager shall set up the program.

[0033]The interrupt signal generation interval (TM time) of the prescribed interval of the timer 15 of the server S is set up at the time of this initialization. The server S asks the run state of that program to a terminal unit while executing the program of an administration object for every TM time of this.

[0034]Nextsecurity check processing of the program in this software management system is explained. A terminal unit by the initialization processing

at the time of program starting when starting execution of a desired program. It sends via LAN10 by making into security check information the login packet (login packet) which stored the identification number of a desired program the identifier of a self-device the user-identification name and the user's password to the server S and an execution demand of the program is performed.

[0035] The server S will judge the validity of the program execution (use) in the terminal unit with reference to a program-management-information table if the login packet of security check information is received from a terminal unit via LAN10.

[0036] Namely the identification number of the program of the request in the login packet received from the server S the identifier of a self-device a user-identification name and the user's password It is judged whether it corresponds to an usable user's concurrency possible number (R) of a program name (N) with the execution demand in a program-management-information table identifier (Tn) of an usable terminal unit identifier (U) and a user's password (PW) If it does not correspond and "execution disapproval" is corresponded execute permission will be returned to a terminal unit as the decision result.

[0037] If the decision result of the validity of the execution to the program which carried out the execution demand is received from the server S a terminal unit It judges whether operation of a program may be started based on it if the decision result received from the server S is "execute permission" execution of a program is started and if it is "execution disapproval" it does not perform by ending the initialization processing of a program. When "execution disapproval" is received from the server S and not executing a program he may leave processing for every program.

[0038] The server S makes a trigger the interrupt signal which the timer 15 generates periodically for every TM time during program execution with each terminal unit A program-management-information table is searched the terminal unit under program execution is searched a request number is attached to the searched terminal unit and the check packet (check packet) which is a packet (Q)



of an inquiry is transmitted to order.

[0039]A terminal unit will return an answer packet (A) if a check packet is received from the server S and a program with an inquiry is performing. When the packet of a reply in predetermined time (TMX time) is not received from the terminal unit of the transmission destination of a check packet the server S judges that the program does not exist at a terminal unit and it does not have a response by causes such as a runaway state and abnormal termination the request number of the program in the terminal unit is made into an initial value "0" and under execution is deleted. This TMX time is good to carry out in several minutes from tens of seconds (for example 60 seconds).

[0040]And a terminal unit transmits the packet (Q) of an inquiry of logout (logout) to the server S in end processing at the time of the end of a program. If the packet is received the server S will make the request number of the program of the applicable terminal unit in the packet an initial value "0" and will delete under execution. Thus a series of processings are completed.

[0041]Drawing 4 is a figure showing an example of a format of the packet of the inquiry exchanged between a terminal unit and a server. The identification information (Q) of the packet of an inquiry is stored in this packet as a header.

[0042]When using it as a login packet in case a terminal unit requires the security check of a program execution demand of this packet from a server as an inquiry of the security check to an execution demand of a program The password of a computer number the identification number which is request number the identification number which is the identifiers of a program the user name which is a user's identifiers and its user is stored as an identifier of a terminal unit.

[0043]Request identification numbers are unique numbers other than "0" attached at the time of login which a terminal unit begins.

It is a number used in order to identify the program on a terminal unit when it sees from the server S and is effective to the logout (logout).

[0044]This request identification number is made to correspond to the identifier

(Tn) of the usable terminal unit shown in drawing 6.

When it is not under execution the initial value "0" is stored and when it is under execution the numerical value sent from the terminal unit is stored.

With the terminal unit which is not an initial value "0" it can judge that a program is performing by searching this request number.

[0045] When the server S uses this packet as a check packet of the inquiry to the terminal unit under program execution as an inquiry under program execution a computer number a request number etc. of a terminal unit of a reference are attached and it transmits to a terminal unit.

[0046] Drawing 5 is a figure showing an example of a format of the packet of a reply exchanged between a terminal unit and a server. The identification information (A) of the packet of a reply is stored in this packet as a header. The packet of a reply to an inquiry of the security check of the program execution demand from a terminal unit and a terminal unit use this packet as a packet [ as opposed to the inquiry from the server under program execution for this packet ] of a reply.

[0047] When the server S uses it as a packet of a reply to an inquiry of the security check of the program execution demand of this packet from a terminal unit as a packet which tells the propriety of program execution The computer number of the terminal unit which asked the security check of the program The request number from the terminal unit the identification number of a program with an execution demand the information that tells permission or the disapproval of execution of the program and the information which shows the reason for disapproval at the time of disapproval are stored.

[0048] Drawing 7 is a figure showing an example of the code of information which tells permission and the disapproval of execution of the program stored in the packet of drawing 5. By the result of a security check the server S stores the code "0" which shows "-; execute permission" when permitting execution of a program to the above-mentioned packet it stores the code "1" which shows "NG; execution disapproval" when execution is disapproval and returns it to a terminal unit.

[0049]Drawing 8 is a figure showing an example of the code of information which tells the reason for disapproval for the program execution stored in the packet of drawing 5. By the result of a security check when carrying out the disapproval of the execution of a program with the code "1" which shows the above-mentioned disapproval the server S also stores the code which shows the reason for disapproval and is returned to a terminal unit.

[0050]as the reason for disapproval -- "-- the code "1" in program identification number unregistered" a computer name (identifier of a terminal unit) -- the code "2" when unregistered In the error which does not expect the code "5" the code "9" is stored when it is over the concurrency possible number and a password is [ code / "3" ] different in a code "4" in the case of an undefined user (users other than an usable user). Although there is a reason for disapproval in addition to this it omits here.

[0051]Drawing 9 is a figure showing the communication procedure of the server at the time of a security check and a terminal unit. First the terminal unit T<sub>n</sub> transmits a login packet (Q) to the server S in the initialization processing at the time of program starting.

[0052]If the login packet (Q) is received the server S performs the security check of the program which had the execution demand from the terminal unit T<sub>n</sub> and if it judges it as execute permission it will return the answer packet (A) which shows it. A terminal unit will start execution of the program if the answer packet of execute permission is received.

[0053]If a login packet (Q) is received from a terminal unit the server S with the interrupt signal for every [ by the timer 15 ] TM time. The check packet (Q) which searches the terminal unit which the program in an administration object is performing and asks the executed situation of a program to the searched terminal unit T<sub>n</sub> is transmitted. And since an answer packet (A) will be returned if a program is that there is nothing normal during execution in the terminal unit T<sub>n</sub> a check is ended in that case.

[0054]After transmitting a check packet (Q) when an answer packet (A) is not

returned within TMX time the request number which shows under execution of the program in the terminal unit is initialized. And a terminal unit will transmit a logout packet (Q) to the server S if a program is ended and the server S will initialize the request number which shows under execution of the program in the terminal unit if the logout packet (Q) is received.

[0055] In the software management system of this example it can perform easily performing maintenance and managing so that the program of software may not be used improperly with each terminal unit using a server. Since a server supervises the executed situation of the program of each terminal units sometimes the runaway state of a program abnormal termination etc. Since the execution right of the program over the terminal unit can be stopped and it can prepare for the execution demand from other terminal units the royalty of a program can be efficiently used with two or more terminal units.

[0056] Next the processing in the case of supervising the idle state of the program in each terminal unit by the server of the software management system of this example is explained. In this case the execution time of the program counted with the counter which it had in the memory storage of the terminal unit  $T_n$  is used. This execution time is the CPU time which the program actually used for example in the case of UNIX OS it is made into "the CPU time which the program used in user's space."

[0057] The server S makes a trigger the interrupt signal periodically generated by the timer 15 for every TM time and transmits the check packet which asks the executed situation of a program to the terminal unit  $T_n$  under program execution for every TM time. If the check packet is received and an applicable program is performing the terminal unit  $T_n$  will store the contents of the counter in an answer packet (A) and will return them to the server S.

[0058] When the server S supervises change of the execution time of the program of the terminal unit based on the contents of the counter stored in the answer packet (A) transmitted from a terminal unit and it is changeless. Although a program is performing it judges that the idle state continues actually and the kill

packet (kill packet) which is a packet which requires the end of the program is transmitted to the terminal unit T<sub>n</sub>. In the terminal unit T<sub>n</sub> if a kill packet is received a positive-acknowledge packet (A) will be returned to the server S and the program of an idle state is terminated.

[0059] Drawing 10 is a figure showing other communication procedures of the server at the time of a security check and a terminal unit. First the terminal unit T<sub>n</sub> transmits a login packet (Q) to the server S in the initialization processing at the time of program starting.

[0060] If the login packet (Q) is received the server S performs the security check of the program which had the execution demand from the terminal unit T<sub>n</sub> and if it judges it as execute permission it will return the answer packet (A) which shows it. A terminal unit will start execution of the program if the answer packet of execute permission is received.

[0061] If a login packet (Q) is received from a terminal unit the server S with the interrupt signal for every [ by the timer 15 ] TM time. The check packet (Q) which searches the terminal unit which the program in an administration object is performing and asks the executed situation of a program to the searched terminal unit T<sub>n</sub> is transmitted.

[0062] The terminal unit T<sub>n</sub> will return the answer packet (A) which stored the contents of the counter to the server S if the check packet (Q) is received. The server S compares with the CPU execution time to which the CPU execution time which is the contents of the counter stored in the answer packet (A) has been returned by last time if an answer packet (A) is returned within TMX time after transmitting a check packet (Q).

[0063] Usually since the CPU execution time stored in the answer packet (A) returned within TMX time at the time of execution is changing from CPU execution time when it checks to last time it admits continuation of execution of a program. However CPU time when it checks to last time is judged to be an idle state if it is not changing as compared with last time and the kill packet (Q) of a program terminating request is transmitted to the terminal unit.

[0064]If the kill packet (Q) is received a terminal unit will return a positive-acknowledge packet (A) and will terminate the program. The server S will initialize the request number which shows execution of the program of the terminal unit if a positive-acknowledge packet (A) is received from a terminal unit.

[0065]Thus when a server can always supervise the run state of the program on each terminal unit and a program is in an idle state in each terminal unit for a long time. Execution of the program over the terminal unit can be stopped it can prepare for the execution demand from other terminal units and the royalty of a program can be efficiently used with two or more terminal units.

[0066]Next the mediation processing of the execution right of a program to each terminal unit by the server of the software management system of this example is explained. When the server S receives the execution demand exceeding the concurrency possible number beforehand decided from the terminal unit T<sub>n</sub> to the program When an execution demand new when the program is executed with the concurrency possible number decided beforehand is received the execution demand is arbitrated with the priority set up beforehand.

[0067]That is when a login packet is received from a terminal unit the server S performs the security check about a program with the execution demand by the packet and are an usable terminal unit and a user.

When a concurrency possible number still has a margin permission of program execution is immediately given to the terminal unit.

[0068]When there is no margin in a concurrency possible number With reference to the program execution priority of a program-management-information table the thing under program execution is searched with the terminal unit whose priority is lower than the terminal unit Kill (kill) of the program execution to the searched terminal unit is carried out it replaces with it and an execution right is permitted to a terminal unit with a high priority.

[0069]Drawing 11 is a figure showing the communication procedure of further others of the server at the time of a security check and a terminal unit. Here the

case where the right of program execution of the terminal unit T1 which uses the program name "spreadsheet C" of the program-management-information table shown in drawing 6 T2 T3 T4 and T5 is arbitrated is explained. The concurrency possible number of this program name "spreadsheet C" is "4."

[0070] First the server S permits execution of the program of "the spreadsheet C" to the terminal unit T1 T2 and T4 and the program is under execution in the terminal unit T1 T2 and T4 respectively.

[0071] If the login packet (Q) of an execution demand of the program of the "spreadsheet C" of a program identifier "P06" is received from the terminal unit T5 the server S Since it checks that they are an usable terminal unit and an usable user and a concurrency possible number has a margin the answer packet (A) which permits the execution right of a program "spreadsheet C" to the terminal unit T5 is sent. In the terminal unit T5 the answer packet (A) is received and execution of a program "spreadsheet C" is started.

[0072] The server S asks and supervises the executed situation of a program for every TM time after receiving each login packet (Q) to each terminal unit T1 T2 T4 and T5.

[0073] In that case when there is no response within TMX time it is judged as a reckless run or abnormal termination of a program and the request number of a program-management-information table is initialized and a program execution possible number is secured. The CPU time stored in the answer packet (A) from each terminal unit will be made to continue as it is if changeful as compared with last time but if changeless it will be judged as an idle state and a program is terminated to the terminal unit and a program execution possible number is secured.

[0074] Then if the login packet (Q) of an execution demand of the program of the "spreadsheet C" of a program identifier "P06" is received from terminal unit T3 the server S Since there is no margin in a concurrency possible number after checking that they are an usable terminal unit and an usable user with reference to a program execution priority the terminal unit under execution is searched for



the present program "spreadsheet C" with a priority lower than terminal unit T3.

[0075]In this case since the terminal unit T5 corresponds to it the server S transmits a kill packet (Q) to the terminal unit T5. Performing the deactivate request of operation of a program "spreadsheet C" the terminal unit T5 ends operation of a program "spreadsheet C" based on the kill packet (Q) snatches an execution right returns an affirmation reply (A) to the server S and returns an execution right.

[0076]If an affirmation reply (A) is received from the terminal unit T5 the server S will initialize the request number of the terminal unit T5 of a program-management-information table will store the request number received from terminal unit T3 and will transmit the answer packet (A) of execute permission to terminal unit T3. If an answer packet (A) is received from the server S terminal unit T3 will give the execution right of a program "spreadsheet C" and will perform the program.

[0077]Also when there is an execution demand of the same program from each terminal units T1-T5 for example Execute permission can be notified to the priority order of each terminal units T1-T5 to a concurrency possible number execution disapproval can be notified to the terminal unit exceeding a concurrency possible number and competition of an execution demand of two or more terminal units can also be arbitrated.

[0078]Although the above-mentioned example explained the case where it performed only when the program is beforehand stored in each terminal unit and there is a notice of allowance from the server S The program which each terminal unit shares is stored in the server S and there is an execution demand from each terminal unit and the execution is permitted and it may be made to enable use of a program.

[0079]In the program in which the program of initialization processing is also set to main programs such as a word processor and a spreadsheet when there is an execution demand of a main program at the time of the initialization processing what is necessary is just made to perform above-mentioned

processing.

[0080]The password which only the identifier of a program or an identification numberthe concurrency possible number of a programa terminal unit name or a terminal numbera user-identification name or a user identification numberand a user can know is usedconsidering it as the item which judges the validity of execution of a program by the security check of a program -- every program -- beforehand -- a conclusive factor -- \*\*\*\* -- since execute permission is given only when in agreement with conditionsthe unauthorized use of a program can be prevented more effectively.

[0081]If the identifier and identification number are made unique on a networkWhen applying to the system which connected two or more networksthe item which judges the validity of execution of a program by the security check of a program for every network can be changedThere is nothing about a possibility of giving permission of program execution to the terminal unit of an accidentally different systemand the unauthorized use of the program in a wide area network can be prevented.

[0082]Thuswhen the execution demand for which a server exceeds the concurrency possible number decided beforehand is receivedWhen an execution demand new when the concurrency possible number decided beforehand is performed is receivedAnd since the execution right of a program can be preferentially permitted to a terminal unit with a high priority when an execution demand of the program from the terminal unit exceeding a concurrency possible number competesit is not necessary to terminate an urgent execution demand. Thereforethe execution right of a program can be arbitrated according to a user's demandand the royalty of a program can be efficiently used with two or more terminal units.

[0083]The software management system of this example prevents the illegal copy and unauthorized use of a programand there can be no trouble in execution of a program at each terminal unit on a computer network system.

[0084]In order that a server may always supervise the run state of the program

on each terminal unit On a terminal unit a program regains an execution right to a server when abnormal termination is carried out a runaway state and Also when it can prepare for the execution demand from other terminal units and a program is in an idle state on a terminal unit for a long time an execution right can be regained to a server and it can prepare for the execution demand from other terminal units.

[0085] Since a server arbitrates the execution right of the program over each terminal unit based on the priority set as each terminal unit the right of program execution according to a user can be given. Therefore the unauthorized use of the program on a network can be prevented and the royalty of a program can be used efficiently using each terminal unit on a network effectively.

[0086]

[Effect of the Invention] As explained above according to the software management system by this invention maintenance and management can be easily performed so that the program of software may not be used improperly with two or more terminal units connected by means of communications such as LAN.

---

## DESCRIPTION OF DRAWINGS

---

[Brief Description of the Drawings]

[Drawing 1] It is a block diagram showing the composition of the software management system which is one example of this invention.

[Drawing 2] It is a block diagram showing the composition of each terminal units T1-Tn shown in drawing 1.

[Drawing 3] It is a block diagram showing the composition of the server S shown in drawing 1.

[Drawing 4] It is a figure showing an example of a format of the packet of the inquiry exchanged between the terminal unit and server which were shown in

drawing 1.

[Drawing 5] It is a figure showing an example of a format of the packet of a reply exchanged between the terminal unit and server which were shown in drawing 1.

[Drawing 6] It is a figure showing an example of a format of the program-management-information table stored in the memory storage 12 of the server S shown in drawing 3.

[Drawing 7] It is a figure showing an example of the code of information which tells permission and the disapproval of execution of the program stored in the packet of drawing 5.

[Drawing 8] It is a figure showing an example of the code of information which tells the reason for disapproval for the program execution stored in the packet of drawing 5.

[Drawing 9] It is a figure showing the communication procedure of the server at the time of the security check in the software management system shown in drawing 1 and a terminal unit.

[Drawing 10] It is a figure showing other communication procedures of the server at the time of the security check in the software management system shown in drawing 1 and a terminal unit.

[Drawing 11] It is a figure showing the communication procedure of further others of the server at the time of the security check in the software management system shown in drawing 1 and a terminal unit.

[Description of Notations]

111: Input/output device 212: Memory storage

313: LAN interface

14[ 4 and ]:CPU 15: Timer

16: Management table counter

10: Local Area Network (LAN)

T1-Tn: Terminal unit S: Server

---